SUTTER INSTRUMENT®

Lambda 721 External Control Quick Reference

Rev. 1.10c (20210517) (FW v1.26+)

Controlling the Lambda 721 externally via computer is accomplished by sending commands over the USB interface between the computer and the relevant connector on the rear panel of the Lambda 721 controller.

USB Interface: The USB device driver for Windows is downloadable from Sutter Instrument's web site (www.sutter.com). The Lambda 721 requires USB CDM (Combined Driver Model) Version 2.10.00 or higher. The CDM device driver for the Lambda 721 consists of two device drivers: 1) USB device driver, and 2) VCP (Virtual COM Port) device driver. Install the USB device driver first, followed by the VCP device driver. The VCP device driver provides a serial RS-232 I/O interface between a Windows application and the Lambda 721. Although the VCP device driver is optional, its installation is recommended even if it is not going to be used. Once installed, the VCP can be enabled or disabled.

The CDM device driver package provides two I/O methodologies over which communications with the Lambda 721 can be conducted: 1). USB Direct, or 2). Serial RS-232 asynchronous via the VCP device driver. The first method requires that the VCP device driver be disabled (or not installed). The second method requires that the VCP be installed and enabled.

Virtual COM Port (VCP) Serial Port Settings: The following table lists the required RS-232 serial settings for the COM port (COM3, COM5, etc.) generated by the installation of the VCP device driver.

Table 1. Serial port settings.

Property	Setting
Data ("Baud") Rate (bits per second (bps)) (Configurable with DIP Switch 5.)	**9600***, 57600
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

** Factory default setting.*

The settings shown in the above table can be set in the device driver's properties (via the Device Manager if in Windows) and/or programmatically in your application.

Handshaking: Command sequences do not have terminators. If a command sequence just sent to the controller is determined to be valid, the entire sequence is immediately echoed back to the computer. All commands return an ASCII CR (Carriage Return; 13 decimal, 0D hexadecimal) to indicate that the task associated with the command has completed. When the Lambda 721 completes the task associated with a command, it sends an ASCII CR back to the host computer indicating that it is ready to receive a new command. If a command returns data, the last byte returned is the task-completed indicator.

Commands: Each command sequence consists of at least one byte, the first of which is the "command byte". Those commands that have parameters or arguments require a sequence of bytes that follow the command byte. No delimiters are used between command sequence arguments. Every command and command sequence ends with a terminator byte containing an ASCII CR (13 decimal, 0D hexadecimal). Although most command bytes can be expressed as ASCII displayable/printable characters, the rest of a command sequence must always be expressed as a sequence of unsigned byte values (0-255 decimal; 00 – FF hexadecimal, or 00000000 – 11111111 binary). Each byte in a command sequence being transmitted to the controller must contain an unsigned binary value. Attempting to code command sequences as "strings" is not advisable. Any command

data being returned from the controller must also be received and initially treated as a sequence of unsigned byte values. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit "word" or "short", or 4 bytes into a 32-bit "long" or "double word"). For the Lambda 721, all Ring Buffer entry values are stored as "unsigned short" (16-bit) values. A 16-bit value is transmitted and received to and from the controller as two contiguous bytes.

"Unsigned" means the value can only be positive; negative values are not permitted. A U16 consists of two contiguous bytes, with a byte/bit-ordering format of Little Endian ("Intel") (most significant byte (MSB) in the first byte and least significant (LSB) in the last byte). If the platform on which your application is running is Little Endian, then no byte order reversal of Ring Buffer entry values is necessary. Examples of platforms using Little Endian formatting include any system using an Intel processor (including Microsoft Windows and Apple Mac OS X), and most Linux distributions running on Intel/AMD processor-based systems.

If the platform on which your application is running is "Big Endian" ("Motorola"), then these U16 position values must have their bytes reverse-ordered after receiving from, or before sending to, the Lambda 721. Examples of Big-Endian platforms include most all non-Intel-based systems, LabVIEW (regardless of system & operating system), and Java (programming language/environment).

Command Reference: The following tables lists all the external-control commands for the Lambda 721.

Table 2. Lambda 721 external control commands.

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
Lambda 10-x Mode ('L' or 'l')	**Tx**	All	1	0	76 or 108	4C or 63	0100 1100 or 0110 1100	0076 or 0108		'L' or 'l'	Places Lambda 721 into Lambda 10-x compatibility mode.
	Rx	All	1	12	13	0D	0000 1101			<CR>	Completion indicator
TTL Mode ('T' or 't')	**Tx**	All	1	0	84 or 116	54 or 74	0101 0100 or 0111 0100	0084 or 0116		'T' or 't'	Places Lambda 721 into TTL mode.
	Rx	All	1	12	13	0D	0000 1101			<CR>	Completion indicator
Start Ring Buffer Run ('R' or 'r')	**Tx**	All	1	0	82 or 114	52 or 72	0101 0010 or 0111 0010	0082 or 0114		'R' or 'r'	Start running the ring buffer.
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator
Stop TTL or Ring Buffer Run ('O' or 'o')	**Tx**	All	1	0	79 or 111	4F or 6F	0100 1111 or 0110 1111	0079 or 0111		'O' or 'o'	Stop TTL Mode or Buffer Ring run.
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator
Load Ring Buffer ('B' or 'b')	**Tx**	All	1	0	66 or 98	42 or 62	0100 0010 or 0110 0010	0066 or 0098		'B' or 'b'	Begin loading the ring buffer.
			2 – 200	1 – 199	Each two-byte ("word") value that follows is the entry for the position, and what follows after is for the next entry (see the *Ring Buffer Entry Values* table), until F0F0 hexadecimal is sent, which stops the loading of the ring buffer. Up to 100 entries are supported.						
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator is returned after the Stop Loading Ring Buffer entry (see next) is sent, effectively ending the command

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
											sequence.
Turn on a discrete LED or a group of LEDs ('M' or 'm')	**Tx**	All	2	0	77 or 109	4D or 6D	01001101 or 01101101	0077 or 0109		'M' or 'm'	Command followed by 1 byte containing the bit-encoded ON/OFF states of all LEDs (1 – 7).
				1	0 – 127	00 – 7F	00000000 – 01111111	0000 – 0127	^@ –	<NUL> – <DEL>	See *'M' Command 1st argument (2nd byte) values for individual LED ON/OFF state bit encoding* for a discrete LED value or *'M' Command 1st argument (2nd byte) values for ON/OFF state bit encoding for all LEDs and groups* for an LED group value.
	Rx	All	1	1	13	0D	0000 1101		^M	<CR>	Completion indicator
Set LED Power Level ('P' or 'p')	**Tx**	All	3	0	80 or 112	50 or 70	0101 0000 or 0111 0000	0080 or 0112		'P' or 'p'	Sets the power level (0 – 100%) for a specified LED.
				1	1 – 7	01 – 07	0000 0001 – 0000 0111	0001 – 0007	^A – ^G	<SOH> – <BEL>	LED number (1 – 7)
				2	1 – 100	01 – 64	0000 0001 – 0110 0100	0001 – 0100	^A –	<SOH> – 'd'	Power level (1 – 100%)
	Rx	All	3	1	1 – 7	01 – 07	0000 0001 – 0000 0111	0001 – 0007		<SOH> – <BEL>	Echoed LED number (1 – 7)
				2	1 – 100	01 – 64	0000 0001 – 0110 0100	0001 – 0100		<SOH> – 'd'	Echoed power level (1 – 100%)
				0	13	0D	0000 1101			<CR>	Completion indicator
Get LED ON/OFF Status ('S' or 's')	**Tx**	All	1	0	83 or 115	53 or 73	0101 0011 or 0111 0011	0083 or 0115		'S' or 's'	Returns a value of 0 (all LEDs off), or one or more ASCII digits ('1' – '7') for each LED that is ON.
	Rx	All	2 - 8	0	0 or 49 – 55	00 or 31 – 37	0000 0000 or 0011 0001 – 0011 0111			<NUL> or '1' – '7'	Value 0 for all LEDs OFF, or ASCII digit '1' – '7' for each LED that is ON, in that order (1 to 7 bytes).
				1 - 8	13	0D	0000 1101			<CR>	Completion indicator
Get Lambda 10-3 Compatible Controller Type and Configuration (see note)	**Tx**	1.26+	1	0	253	FD	1111 1101	0253			Returns a Lambda 10-3 compatible data block containing controller type identifier, and filter wheel /shutter configuration information. (See note.)
	Rx	1.26+	31	0	253	FD	1111 1101				Command echo
				1 (4)	49	31	0011 0001			1	"10-3" (controller type is Lambda 10-3)
					48	30	0011 0000			0	
					45	2D	0010 1101			-	
					51	33	0011 0011			3	
				5 (5)	87	57	0101 0111			W	"WA-25" (Filter Wheel A is a 10-position wheel with 25mm filters)
					65	41	0100 0001			A	
					45	2D	0010 1101			-	
					50	32	0011 0010			2	
					53	35	0011 0101			5	

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
				10 (5)	87	57	0101 0111			W	"WB-NC" (Filter Wheel B is not connected)
					66	42	0100 0010			B	
					45	2D	0010 1101			-	
					78	4E	0100 1110			N	
					67	43	0100 0011			C	
				15 (5)	87	57	0101 0111			W	"WC-NC" (Filter Wheel C is not connected)
					67	43	0100 0011			C	
					45	2D	0010 1101			-	
					78	4E	0100 1110			N	
					67	43	0100 0011			C	
				20 (5)	83	53	0101 0011			S	"SA-VS" (Shutter A is not connected or is a standard shutter (not a SmartShutter))
					65	41	0100 0001			A	
					45	2D	0010 1101			-	
					86	56	0101 0110			V	
					83	53	0101 0011			S	
				25 (5)	83	53	0101 0011			S	"SB-VS" (Shutter B is not connected or is a standard shutter (not a SmartShutter))
					66	42	0100 0010			B	
					45	2D	0010 1101			-	
					86	56	0101 0110			V	
					83	53	0101 0011			S	
				30	13	0D	0000 1101			<CR>	Completion indicator
Get Lambda 10-3 Compatible Status (see note)	**Tx**	1.26+	1	0	204	CC	1100 1100	0204			Returns a Lambda 10-3 compatible data block containing filter wheel and shutter status information. (See note.)
	Rx	1.26+	13	0	204	CC	1100 1100				Command echo
				1	16	10	0001 0000				Wheel A, Speed 1, Pos. 0
				2	138	8A	1000 1010				Wheel B, Speed 0, Pos. 10
				3 (2)	252	FC	1111 1100				Wheel C prefix byte
					10	0A	0000 1010				Wheel C, Speed 0, Pos. 10
				5	172	AC	1010 1100				Shutter A closed
				6	188	BC	1011 1100				Shutter B closed
				7 (2)	219	DB	1101 1011				Shutter A mode: N/A
					1	01	0000 0001				Shutter A designator
				9 (2)	219	DB	1101 1011				Shutter B mode: N/A
					2	02	0000 0010				Shutter B designator
				11	13	0D	0000 1101			<CR>	Completion indicator
				12	13	0D	0000 1101			<CR>	Completion indicator

***NOTE**: The "**Get Lambda 10-3 Compatible Controller Type and Configuration**" and "**Get Lambda 10-3 Compatible Status**" commands are provided to allow external-control software originally written for the Lambda 10-3 to control a Lambda 721 while identifying itself as a Lambda 10-3. All data returned is static (i.e., data remains unchanged regardless of all Lambda 721 states). These two commands are functional only if DIP Switch 2 is OFF (up).*

Table 3. Lambda 10 Series compatible LED selection commands.

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
All LEDs Off	**Tx**	All	1	0	0	00	00000000	0000	^@	<NUL>	Turn all LEDs off.
	Rx	All	2	0	0	00	00000000		^@	<NUL>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 1 On	**Tx**	All	1	0	1	01	00000001	0001	^A	<SOH>	Turn LED 1 on
	Rx	All	2	0	1	01	00000001		^A	<SOH>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 2 On	**Tx**	All	1	0	2	02	00000010	0002	^B	<STX>	Turn LED 2 on
	Rx	All	2	0	2	02	00000010		^B	<STX>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 3 On	**Tx**	All	1	0	3	03	00000011	0003	^C	<ETX>	Turn LED 3 on
	Rx	All	2	0	3	03	00000011		^C	<ETX>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 4 On	**Tx**	All	1	0	4	04	00000100	0004	^D	<EOT>	Turn LED 4 on
	Rx	All	2	0	4	04	00000100		^D	<EOT>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 5 On	**Tx**	All	1	0	5	05	00000101	0005	^E	<ENQ>	Turn LED 5 on
	Rx	All	2	0	5	05	00000101		^E	<ENQ>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 6 On	**Tx**	All	1	0	6	06	00000110	0006	^F	<ACK>	Turn LED 6 on
	Rx	All	2	0	6	06	00000110		^F	<ACK>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 7 On	**Tx**	All	1	0	7	07	00000111	0007	^G	<BEL>	Turn LED 7 on
	Rx	All	2	0	7	07	00000111		^G	<BEL>	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator

NOTE: *All Lambda 10 Mode commands for the Lambda 721 correspond to the Lambda 10 series filter wheel movement for Wheel A, Speed 0, Positions 1 – 7. Any other wheel-select, speed settings, and positions 8 – 9 are not supported. Values 1 through 7 correspond to LEDs 1 – 7. Value 0, used to turn all LEDs off, can be used as the equivalent of closing Shutter A.*

Table 4. LED selection commands using ASCII digits.

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
All LEDs Off ('0')	**Tx**	All	1	0	48	30	00110000	0048		0	Turn all LEDs off.
	Rx	All	2	0	48	30	00110000	0048		0	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 1 On ('1')	**Tx**	All	1	0	49	31	00110001	0049		1	Turn LED 1 on
	Rx	All	2	0	49	31	00110001	0049		1	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 2 On ('2')	**Tx**	All	1	0	50	32	00110010	0050		2	Turn LED 2 on
	Rx	All	2	0	50	32	00110010	0050		2	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 3 On ('3')	**Tx**	All	1	0	51	33	00110011	0051		3	Turn LED 3 on
	Rx	All	2	0	51	33	00110011	0051		3	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 4 On ('4')	**Tx**	All	1	0	52	34	00110100	0052		4	Turn LED 4 on
	Rx	All	2	0	52	34	00110100	0052		4	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 5 On ('5')	**Tx**	All	1	0	53	35	00110101	0053		5	Turn LED 5 on
	Rx	All	2	0	53	35	00110101	0053		5	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 6 On ('6')	**Tx**	All	1	0	54	36	00110110	0054		6	Turn LED 6 on
	Rx	All	2	0	54	36	00110110	0054		6	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator
LED 7 On ('7')	**Tx**	All	1	0	55	37	00110111	0055		7	Turn LED 7 on
	Rx	All	2	0	55	37	00110111	0055		7	Command echo
				1	13	0D	0000 1101		^M	<CR>	Completion indicator

Table 5. Ring Buffer entry values.

LED On/Off State	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	16-bit "Word" value (Little Endian)		
					Dec.	Hex.	Binary				Dec.	Hex.	Binary
All Off	Tx	All	2	0	0	00	0000 0000	0000	^@	<NUL>	8	0008	00000000 00001000
				1	8	08	0000 1000	0008	^H	<BS>			
LED 1 On	Tx	All	2	0	1	01	0000 0001	0001	^A	<SOH>	272	0110	00000001 00010000
				1	16	10	0001 0000	0016	^P	<DLE>			
LED 2 On	Tx	All	2	0	2	02	0000 0010	0002	^B	<STX>	536	0218	00000010 00011000
				1	24	18	0001 1000	0024	^X	<CAN>			
LED 3 On	Tx	All	2	0	4	04	0000 0100	0004	^D	<EOT>	1056	0420	00000100 00100000
				1	32	20	0010 0000	0032		space			
LED 4 On	Tx	All	2	0	8	08	0000 1000	0008	^H	<BS>	2088	0828	00001000 00101000
				1	40	28	0010 1000	0040		(			
LED 5 On	Tx	All	2	0	16	10	0001 0000	0016	^P	<DLE>	4144	1030	00010000 00110000
				1	48	30	0011 0000	0048		0			
LED 6 On	Tx	All	2	0	32	20	0010 0000	0032		space	8248	2038	00100000 00111000
				1	56	38	0011 1000	0056		8			
LED 7 On	Tx	All	2	0	64	40	0100 0000	0064		@	16448	4040	01000000 01000000
				1	64	40	0100 0000	0064		@			
End of Ring Buffer	Tx	All	2	0	240	F0	1111 0000	0240		≡	61680	F0F0	11110000 11110000
				1	240	F0	1111 0000	0240		≡			
	Rx	All	1	1	13	0D	0000 1101		^M	<CR>	Completion indicator		

NOTE: *The 'B' or 'b' command and each 2-byte Ring Buffer entry, once transmitted, does not return anything, except for End of Ring Buffer which returns an ASCII CR (carriage return) completion indicator (13 decimal, 0D hexadecimal, 0000 1101 binary).*

Table 6. Return values while running the Ring Buffer.

LED On/Off State	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			ASCII def./-char.
					Dec.	Hex.	Binary	
LED 1 On	Rx	All	1	0	49	31	0011 0001	'1'
LED 2 On	Rx	All	1	0	50	32	0011 0010	'2'
LED 3 On	Rx	All	1	0	51	33	0011 0011	'3'
LED 4 On	Rx	All	1	0	52	34	0011 0100	'4'
LED 5 On	Rx	All	1	0	53	35	0011 0101	'5'
LED 6 On	Rx	All	1	0	54	36	0011 0110	'6'
LED 7 On	Rx	All	1	0	55	37	0011 0111	'7'

NOTE: *The return of the values shown in the table while running the ring buffer is contingent on DIP Switch 4 being set to the ON (down) position.*

Table 7. 'M' Command 1st argument (2nd byte) values for individual LED ON/OFF state bit encoding.

LED # On/Off State (Bit #)								Command 'M' or 'm' Byte Argument	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.
n/a (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)		Dec.	Hex.	Binary			
								All LEDs Off	0	00	00000000	0000	^@	<NUL>
							ON	LED 1 On	1	01	00000001	0001	^A	<SOH>
						ON		LED 2 On	2	02	00000010	0002	^B	<STX>
					ON			LED 3 On	4	04	00000100	0004	^D	<EOT>
				ON				LED 4 On	8	08	00001000	0008	^H	<BS>
			ON					LED 5 On	16	10	00010000	0016	^P	<DLE>
		ON						LED 6 On	32	20	00100000	0032		<DEL>
	ON							LED 7 On	64	40	01000000	0064		@

Table 8. 'M' Command 1st argument (2nd byte) values for ON/OFF state bit encoding for all LEDs and groups.

LED # On/Off State (Bit #)								Command 'M' or 'm' Byte Argument	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.
n/a (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)		Dec.	Hex.	Binary			
								All LEDs Off	0	00	00000000	0000	^@	<NUL>
							ON	LED 1 On	1	01	00000001	0001	^A	<SOH>
						ON		LED 2 On	2	02	00000010	0002	^B	<STX>
						ON	ON	LED 1 2 On	3	03	00000011	0003	^C	<ETX>
					ON			LED 3 On	4	04	00000100	0004	^D	<EOT>
					ON		ON	LED 3 1 On	5	05	00000101	0005	^E	<ENQ>
					ON	ON		LED 3 2 On	6	06	00000110	0006	^F	<ACK>
					ON	ON	ON	LED 3 2 1 On	7	07	00000111	0007	^G	<BEL>
				ON				LED 4 On	8	08	00001000	0008	^H	<BS>
				ON			ON	LED 4 1 On	9	09	00001001	0009	^I	<HT>
				ON		ON		LED 4 2 On	10	0A	00001010	0010	^J	<LF>
				ON		ON	ON	LED 4 2 1 On	11	0B	00001011	0011	^K	<VT>
				ON	ON			LED 4 3 On	12	0C	00001100	0012	^L	<FF>
				ON	ON		ON	LED 4 3 1 On	13	0D	00001101	0013	^M	<CR>
				ON	ON	ON		LED 4 3 2 On	14	0E	00001110	0014	^N	<SO>
				ON	ON	ON	ON	LED 4 3 2 1 On	15	0F	00001111	0015	^O	<SI>
			ON					LED 5 On	16	10	00010000	0016	^P	<DLE>
			ON				ON	LED 5 1 On	17	11	00010001	0017	^Q	<DC1>
			ON			ON		LED 5 2 On	18	12	00010010	0018	^R	<DC2>
			ON			ON	ON	LED 5 2 1 On	19	13	00010011	0019	^S	<DC3>
			ON		ON			LED 5 3 On	20	14	00010100	0020	^T	<DC4>
			ON		ON		ON	LED 5 3 1 On	21	15	00010101	0021	^U	<NAK>
			ON		ON	ON		LED 5 3 2 On	22	16	00010110	0022	^V	<SYN>
			ON		ON	ON	ON	LED 5 3 2 1 On	23	17	00010111	0023	^W	<ETB>
			ON	ON				LED 5 4 On	24	18	00011000	0024	^X	<CAN>
			ON	ON			ON	LED 5 4 1 On	25	19	00011001	0025	^Y	<EM>
			ON	ON		ON		LED 5 4 2 On	26	1A	00011010	0026	^Z	<SUB>
			ON	ON		ON	ON	LED 5 4 2 1 On	27	1B	00011011	0027	^[	<ESC>
			ON	ON	ON			LED 5 4 3 On	28	1C	00011100	0028	^\	<FS>
			ON	ON	ON		ON	LED 5 4 3 1 On	29	1D	00011101	0029	^]	<GS>

LED # On/Off State (Bit #)								Command 'M' or 'm' Byte Argument	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.
n/a (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)		Dec.	Hex.	Binary			
			ON	ON	ON	ON		**LED 5 4 3 2 On**	30	1E	00011110	0030	^^	<RS>
			ON	ON	ON	ON	ON	**LED 5 4 3 2 1 On**	31	1F	00011111	0031	^_	<US>
		ON						**LED 6 On**	32	20	00100000	0032		<DEL>
		ON					ON	**LED 6 1 On**	33	21	00100001	0033		!
		ON				ON		**LED 6 2 On**	34	22	00100010	0034		"
		ON				ON	ON	**LED 6 1 2 On**	35	23	00100011	0035		#
		ON			ON			**LED 6 3 On**	36	24	00100100	0036		$
		ON			ON		ON	**LED 6 3 1 On**	37	25	00100101	0037		%
		ON			ON	ON		**LED 6 3 2 On**	38	26	00100110	0038		&
		ON			ON	ON	ON	**LED 6 3 2 1 On**	39	27	00100111	0039		'
		ON		ON				**LED 6 4 On**	40	28	00101000	0040		(
		ON		ON			ON	**LED 6 4 1 On**	41	29	00101001	0041		)
		ON		ON		ON		**LED 6 4 2 On**	42	2A	00101010	0042		*
		ON		ON		ON	ON	**LED 6 4 2 1 On**	43	2B	00101011	0043		+
		ON		ON	ON			**LED 6 4 3 On**	44	2C	00101100	0044		,
		ON		ON	ON		ON	**LED 6 4 3 1 On**	45	2D	00101101	0045		-
		ON		ON	ON	ON		**LED 6 4 3 2 On**	46	2E	00101110	0046		.
		ON		ON	ON	ON	ON	**LED 6 4 3 2 1 On**	47	2F	00101111	0047		/
		ON	ON					**LED 6 5 On**	48	30	00110000	0048		0
		ON	ON				ON	**LED 6 5 1 On**	49	31	00110001	0049		1
		ON	ON			ON		**LED 6 5 2 On**	50	32	00110010	0050		2
		ON	ON			ON	ON	**LED 6 5 2 1 On**	51	33	00110011	0051		3
		ON	ON		ON			**LED 6 5 3 On**	52	34	00110100	0052		4
		ON	ON		ON		ON	**LED 6 5 3 1 On**	53	35	00110101	0053		5
		ON	ON		ON	ON		**LED 6 5 3 2 On**	54	36	00110110	0054		6
		ON	ON		ON	ON	ON	**LED 6 5 3 2 1 On**	55	37	00110111	0055		7
		ON	ON	ON				**LED 6 5 4 On**	56	38	00111000	0056		8
		ON	ON	ON			ON	**LED 6 5 4 1 On**	57	39	00111001	0057		9
		ON	ON	ON		ON		**LED 6 5 4 2 On**	58	3A	00111010	0058		:
		ON	ON	ON		ON	ON	**LED 6 5 4 2 1 On**	59	3B	00111011	0059		;
		ON	ON	ON	ON			**LED 6 5 4 3 On**	60	3C	00111100	0060		<
		ON	ON	ON	ON		ON	**LED 6 5 4 3 1 On**	61	3D	00111101	0061		=
		ON	ON	ON	ON	ON		**LED 6 5 4 3 2 On**	62	3E	00111110	0062		>
		ON	ON	ON	ON	ON	ON	**LED 6 5 4 3 2 1 On**	63	3F	00111111	0063		?
	ON							**LED 7 On**	64	40	01000000	0064		@
	ON						ON	**LED 7 1 On**	65	41	01000001	0065		A
	ON					ON		**LED 7 2 On**	66	42	01000010	0066		B
	ON					ON	ON	**LED 7 1 2 On**	67	43	01000011	0067		C
	ON				ON			**LED 7 3 On**	68	44	01000100	0068		D
	ON				ON		ON	**LED 7 3 1 On**	69	45	01000101	0069		E
	ON				ON	ON		**LED 7 3 2 On**	70	46	01000110	0070		F
	ON				ON	ON	ON	**LED 7 3 2 1 On**	71	47	01000111	0071		G
	ON			ON				**LED 7 4 On**	72	48	01001000	0072		H
	ON			ON			ON	**LED 7 4 1 On**	73	49	01001001	0073		I
	ON			ON		ON		**LED 7 4 2 On**	74	4A	01001010	0074		J
	ON			ON		ON	ON	**LED 7 4 2 1 On**	75	4B	01001011	0075		K

LED # On/Off State (Bit #)								Command 'M' or 'm' Byte Argument	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.
n/a (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)		Dec.	Hex.	Binary			
	ON			ON	ON			**LED 7 4 3 On**	76	4C	01001100	0076		L
	ON			ON	ON		ON	**LED 7 4 3 1 On**	77	4D	01001101	0077		M
	ON			ON	ON	ON		**LED 7 4 3 2 On**	78	4E	01001110	0078		N
	ON			ON	ON	ON	ON	**LED 7 4 3 2 1 On**	79	4F	01001111	0079		O
	ON		ON					**LED 7 5 On**	80	50	01010000	0080		P
	ON		ON				ON	**LED 7 5 1 On**	81	51	01010001	0081		Q
	ON		ON			ON		**LED 7 5 2 On**	82	52	01010010	0082		R
	ON		ON			ON	ON	**LED 7 5 2 1 On**	83	53	01010011	0083		S
	ON		ON		ON			**LED 7 5 3 On**	84	54	01010100	0084		T
	ON		ON		ON		ON	**LED 7 5 3 1 On**	85	55	01010101	0085		U
	ON		ON		ON	ON		**LED 7 5 3 2 On**	86	56	01010110	0086		V
	ON		ON		ON	ON	ON	**LED 7 5 3 2 1 On**	87	57	01010111	0087		W
	ON		ON	ON				**LED 7 5 4 On**	88	58	01011000	0088		X
	ON		ON	ON			ON	**LED 7 5 4 1 On**	89	59	01011001	0089		Y
	ON		ON	ON		ON		**LED 7 5 4 2 On**	90	5A	01011010	0090		Z
	ON		ON	ON		ON	ON	**LED 7 5 4 2 1 On**	91	5B	01011011	0091		[
	ON		ON	ON	ON			**LED 7 5 4 3 On**	92	5C	01011100	0092		\
	ON		ON	ON	ON		ON	**LED 7 5 4 3 1 On**	93	5D	01011101	0093		]
	ON		ON	ON	ON	ON		**LED 7 5 4 3 2 On**	94	5E	01011110	0094		^
	ON		ON	ON	ON	ON	ON	**LED 7 5 4 3 2 1 On**	95	5F	01011111	0095		_
	ON	ON						**LED 7 6 On**	96	60	01100000	0096		`
	ON	ON					ON	**LED 7 6 1 On**	97	61	01100001	0097		a
	ON	ON				ON		**LED 7 6 2 On**	98	62	01100010	0098		b
	ON	ON				ON	ON	**LED 7 6 1 2 On**	99	63	01100011	0099		c
	ON	ON			ON			**LED 7 6 3 On**	100	64	01100100	0100		d
	ON	ON			ON		ON	**LED 7 6 3 1 On**	101	65	01100101	0101		e
	ON	ON			ON	ON		**LED 7 6 3 2 On**	102	66	01100110	0102		f
	ON	ON			ON	ON	ON	**LED 7 6 3 2 1 On**	103	67	01100111	0103		g
	ON	ON		ON				**LED 7 6 4 On**	104	68	01101000	0104		h
	ON	ON		ON			ON	**LED 7 6 4 1 On**	105	69	01101001	0105		i
	ON	ON		ON		ON		**LED 7 6 4 2 On**	106	6A	01101010	0106		j
	ON	ON		ON		ON	ON	**LED 7 6 4 2 1 On**	107	6B	01101011	0107		k
	ON	ON		ON	ON			**LED 7 6 4 3 On**	108	6C	01101100	0108		l
	ON	ON		ON	ON		ON	**LED 7 6 4 3 1 On**	109	6D	01101101	0109		m
	ON	ON		ON	ON	ON		**LED 7 6 4 3 2 On**	110	6E	01101110	0110		n
	ON	ON		ON	ON	ON	ON	**LED 7 6 4 3 2 1 On**	111	6F	01101111	0111		o
	ON	ON	ON					**LED 7 6 5 On**	112	70	01110000	0112		p
	ON	ON	ON				ON	**LED 7 6 5 1 On**	113	71	01110001	0113		q
	ON	ON	ON			ON		**LED 7 6 5 2 On**	114	72	01110010	0114		r
	ON	ON	ON			ON	ON	**LED 7 6 5 2 1 On**	115	73	01110011	0115		s
	ON	ON	ON		ON			**LED 7 6 5 3 On**	116	74	01110100	0116		t
	ON	ON	ON		ON		ON	**LED 7 6 5 3 1 On**	117	75	01110101	0117		u
	ON	ON	ON		ON	ON		**LED 7 6 5 3 2 On**	118	76	01110110	0118		v
	ON	ON	ON		ON	ON	ON	**LED 7 6 5 3 2 1 On**	119	77	01110111	0119		w
	ON	ON	ON	ON				**LED 7 6 5 4 On**	120	78	01111000	0120		x
	ON	ON	ON	ON			ON	**LED 7 6 5 4 1 On**	121	79	01111001	0121		y

LED # On/Off State (Bit #)								Command 'M' or 'm' Byte Argument	Value			Alt-key-pad #	Ctrl-code	ASCII def./-char.
n/a (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)		Dec.	Hex.	Binary			
	ON	ON	ON	ON		ON		LED 7 6 5 4 2 On	122	7A	01111010	0122		z
	ON	ON	ON	ON		ON	ON	LED 7 6 5 4 2 1 On	123	7B	01111011	0123		{
	ON	ON	ON	ON	ON			LED 7 6 5 4 3 On	124	7C	01111100	0124		|
	ON	ON	ON	ON	ON		ON	LED 7 6 5 4 3 1 On	125	7D	01111101	0125		}
	ON	ON	ON	ON	ON	ON		LED 7 6 5 4 3 2 On	126	7E	01111110	0126		~
	ON	ON	ON	ON	ON	ON	ON	LED 7 6 5 4 3 2 1 On	127	7F	01111111	0127		

NOTES:

1. **Task-Complete Indicator:** All commands will send back to the computer the "Task-Complete Indicator" to signal the command and its associated function in the controller is complete. The indicator consists of one (1) byte containing a value of 13 decimal (0D hexadecimal), and which represents the ASCII CR (Carriage Return).
2. **Intercommand Delay:** A short delay (usually around 2 ms) is recommended between commands (after sending a command sequence and before sending the next command).
3. **Clearing Send/Receive Buffers:** Clearing (purging) the transmit and receive buffers of the I/O port immediately before sending any command is recommended.
4. **16-Bit Value Bit Ordering:** All multibyte values transmitted to, and received from, the controller must be bit/byte-ordered in "Little Endian" format. This means that the least significant bit/byte is last (last to send and last to receive). Byte-order reversal may be required on some platforms. Microsoft Windows, Intel-based Apple Macintosh systems running Mac OS X, and most Intel/AMD processor-based Linux distributions handle byte storage in Little-Endian byte order, so byte reordering is not necessary before converting to/from 16-bit "short" or "word" values. LabVIEW always handles "byte strings" in "Big Endian" byte order irrespective of operating system and CPU, requiring that the two bytes containing a 16-bit value be reverse ordered before/after conversion to/from a multibyte type value (I16, U16, etc.). MATLAB automatically adjusts the endianess of multibyte storage entities to that of the system on which it is running, so explicit byte reordering is generally unnecessary unless the underlying platform is Big Endian. If your development platform does not have built-in Little/Big Endian conversion functions, bit reordering can be accomplished by first swapping positions of the two bytes in each 16-bit value. This method efficiently and quickly changes the bit ordering of any multibyte value between the two Endian formats (if Big Endian, it becomes Little Endian, and if Little Endian, it becomes then Big Endian).

NOTES:

NOTES: